CLAIMS

What is claimed is:

1. A method for controlling concurrency of access to data in a database system, comprising:

(a) partitioning a table in the database system into a plurality of partitions;

(b) receiving a request for access to data;

(c) determining a partition of the plurality of partitions that contains the data;

(d) determining if the data has been committed; and

(e) if so, avoiding locking the partition in response to the request.

2. The method of claim 1, further comprising:

(f) if it cannot be determined whether the data has been committed:

(f1) locking the partition in response to the request; and

(f2) granting access to the partition.

3. The method of claim 2, wherein granting access to the partition includes serializing access to the partition at a lock state.

4. The method of claim 3, wherein locking the partition includes locking the partition at the lock state and serializing access to the partition includes permitting access to the partition for access requests that are compatible with the lock state.

5. The method of claim 4, wherein the access to the data comprises access by a plurality of applications through a single database system.

6. The method of claim 2, wherein the access to the data comprises access by a plurality of database system.

7. The method of claim 1, wherein receiving a request for access includes receiving a request for a record in the table.

8. The method of claim 1, wherein receiving a request for access includes receiving a request to lock the partition.

9. The method of claim 8, wherein the request is a request for a shared lock.

10. The method of claim 8, wherein the request is a request for an exclusive lock.

11. The method of claim 4, further including receiving a lockmax value, accumulating for an application, a number of requests for access to the records in the table by the application, comparing the number of requests with the lockmax value, and, when the number of requests equals the lockmax value, escalating the lock state.

12. A database management system that manages a database system containing data in tables, comprising:

a database system component to partition a table in the database system into a plurality of partitions; and

a data manager that;

receives a request for access to data

a partition of the plurality of partitions that contains the data;

determines if the data has been committed, and

if so, avoids locking the partition in response to the request.

13. The system of claim 12, wherein the data manager further:

obtains a lock on the partition in response to the request, if it cannot be determined whether the data has been committed.

14. The system of claim 13, further comprising a lock manager that:

grants the lock on the partition; and

serializes access to the partition at a lock state.

15. The system of claim 14, wherein the database manager obtains the lock at the lock state and the lock manager serializes access to the partition by granting requests for locks on the partition that are compatible with the lock state.

16. The system of claim 14, wherein the lock manager is coupled to a plurality of database systems.

17. The system of claim 15, wherein the requests for locks represent requests for access to the table from a plurality of applications coupled to the database management system.

18. The system of claim 15, further including a lockmax value, wherein the database manager accumulates for an application a number of requests for access to records in the table by the application, compares the number of requests with the lock max value, and, when the number of requests equals the lockmax value, requests escalation of the lock state.

19. The system of claim 12, wherein a request for access includes a request for a record in the table.

20. The system of claim 12, wherein the request for access includes a request to lock the partition.

21. The system of claim 20, wherein the request is a request for a shared lock.

22. The system of claim 20, wherein the request is a request for an exclusive lock.

23. A computer readable medium with program instructions for controlling concurrency of access to data in a database system, comprising instructions for:

(a) partitioning a table in the database system into a plurality of partitions;

(b) receiving a request for access to data in a partition in the table;

(c) determining if the data has been committed; and

(d) if so, avoiding locking the partition in response to the request.

24. The medium of claim 23, further comprising instructions for:

(e) if the data has not been committed:

(e1) locking the partition in response to the request, and

(e2) granting access to the partition.

25. The medium of claim 24, wherein granting access to the partition includes serializing access to the partition at a lock state.

26. The medium of claim 25, wherein locking the partition includes locking the partition at the lock state and serializing access to the partition includes permitting access to the partition for access requests that are compatible with the lock state.

27. The medium of claim 26, wherein the access to the data comprises access by a plurality of applications through a single database system.

28. The medium of claim 25, wherein the access to the data comprises access by a plurality of database system.

29. The medium of claim 24, wherein receiving a request for access includes receiving a request for a record in the table.

30. The medium method of claim 24, wherein receiving a request for access includes receiving a request to lock the partition.

31. The medium of claim 30, wherein the request is a request for a shared lock.

32. The medium of claim 30, wherein the request is a request for an exclusive lock.

33. The medium of claim 26, further including receiving a lockmax value, accumulating for an application, a number of requests for access to the records in the table by the application, comparing the number of requests with the lockmax value, and, when the number of requests equals the lockmax value, escalating the lock state.